TripleFDS: Triple Feature Disentanglement and Synthesis for Scene Text Editing

Yuchen Bao[1, 2*], **Yiting Wang**[2], **Wenjian Huang**[1], **Haowei Wang**[2], **Shen Chen**[2], **Taiping Yao**[2], **Shouhong Ding**[2], **Jianguo Zhang**[1, 3†]

[1]Research Institute of Trustworthy Autonomous Systems and Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

[2]Tencent Youtu Lab

[3]Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

Abstract

Scene Text Editing (STE) aims to naturally modify text in images while preserving visual consistency, the decisive factors of which can be divided into three parts, *i.e.*, *text style*, *text content*, and *background*. Previous methods have struggled with *incomplete disentanglement of editable attributes*, typically addressing only one aspect—such as editing text content—thus limiting controllability and visual consistency. To overcome these limitations, we propose **TripleFDS**, a novel framework for STE with disentangled modular attributes, and an accompanying dataset called **SCB Synthesis**. **SCB Synthesis** provides robust training data for triple feature disentanglement by utilizing the "SCB Group", a novel construct that combines three attributes per image to generate diverse, disentangled training groups. Leveraging this construct as a basic training unit, **TripleFDS** first disentangles triple features, ensuring semantic accuracy through inter-group contrastive regularization and reducing redundancy through intra-sample multi-feature orthogonality. In the synthesis phase, **TripleFDS** performs feature remapping to prevent "shortcut" phenomena during reconstruction and mitigate potential feature leakage. Trained on 125,000 SCB Groups, **TripleFDS** achieves state-of-the-art image fidelity (SSIM of 44.54) and text accuracy (ACC of 93.58%) on the mainstream STE benchmarks. Besides superior performance, the more flexible editing of **TripleFDS** supports new operations such as style replacement and background transfer.

Code — https://github.com/yusenbao01/TripleFDS

1 Introduction

Scene Text Editing (STE) empowers users with fine-grained control to modify specific text within an image while preserving the invariance of other elements. As exemplified in Fig. 1, this technology is crucial for diverse applications, including digital content creation (e.g., editing documents and posters) and enhancing various downstream tasks such as Scene Text Removal (Zhang et al. 2024a), Optical Character Recognition (OCR) (Fang et al. 2025), and Anti-forgery (Qu et al. 2023a, 2024; Zhao, Chen, and Huang 2021; Roy et al. 2020; Luo et al. 2025).

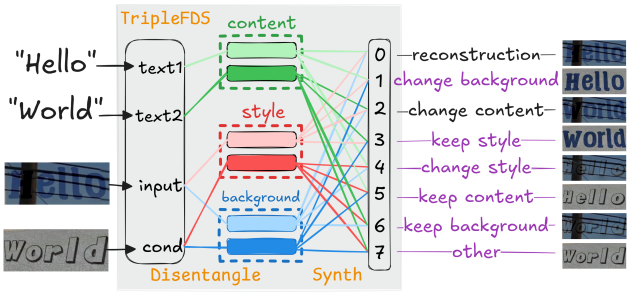

Figure 1: **TripleFDS**'s capabilities. Purple lines denote additional feature permutation-based editing operations enabled by our approach.

Scene Text Editing (STE) methods often employ non-disentangling or implicit strategies, such as diffusion-based inpainting methods (Chen et al. 2023b; Tuo et al. 2024; Chen et al. 2023a) and Transformer-decoder methods (Zhang et al. 2024a; Fang et al. 2025), enabling text reconstruction and inpainting with notable quality. Early attempts at explicit disentanglement, predominantly GAN-based (Gupta, Vedaldi, and Zisserman 2016; Wu et al. 2019; Qu et al. 2023b; Yang, Huang, and Lin 2020; Krishnan et al. 2023; Goodfellow et al. 2014), used distinct modules to separate visual components for tasks like text conversion and style/background extraction. More advanced explicit methods, such as diffusion-based reconstruction paradigms (Zhu et al. 2024; Zeng et al. 2024; Yang et al. 2024), brought enhanced image quality and control by focusing on rich font feature representation and integrating noisy-latent via conditional controls (e.g., ControlNet(Zhang, Rao, and Agrawala 2023) for glyphs(Tuo, Geng, and Bo 2024; Yang et al. 2023), DINOv2(Oquab et al. 2024) for styles(Wang et al. 2024; Liu et al. 2024) through bypass branches or cross-attention mechanisms(Ye et al. 2023; Ji et al. 2024)).

Despite these advances, existing STE methods are primarily limited by *incomplete disentanglement of editable attributes*. For example, RS-STE (Fang et al. 2025)'s insufficient disentanglement often leads to artifacts and blurriness, especially with elaborate fonts or complex backgrounds. Similarly, TextCtrl(Zeng et al. 2024) employs bi-

*Work done during internship at Tencent Youtu Lab.

†Corresponding author.

nary disentanglement, where text content is separated, but style and background remain entangled. This results in inherent stylistic entanglement, causing style deviation and unnatural foreground-background boundaries.

To overcome the significant challenge of *incomplete disentanglement of editable attributes* in previous methods, we propose **TripleFDS**, a novel framework for STE that explicitly disentangles triple features through a dedicated disentanglement and synthesis process. To support this, we introduce **SCB Synthesis**, a novel paradigm for constructing synthetic text image datasets.

To achieve robust triple feature disentanglement, **SCB Synthesis** utilizes the core concept of the SCB Group, which facilitates the natural blending of permutations of text styles, text contents, and backgrounds, generating diverse and well-disentangled training samples. Furthermore, the model can perform self-supervision for features lacking explicit ground truth labels, such as style, by leveraging fixed mapping relationships among samples within a single SCB Group.

Building upon this data construct, we propose two distinct processes for feature disentanglement and synthesis. In the first process, we introduce novel feature disentanglement constraints to achieve accurate and non-redundant feature disentanglement by leveraging the mapping properties both within and across SCB Groups. This strategy enhances feature semantic accuracy, improves feature localization, and enforces orthogonality among features to reduce information redundancy.

In the synthesis process, unlike previous methods that primarily focus on editing, our approach emphasizes robust reconstruction. To achieve this, we introduce a feature remapping strategy that prevents "shortcut" phenomena during reconstruction and mitigates potential feature leakage. Leveraging the fixed feature mapping relationships within the SCB Group, we deliberately create "hard-to-reconstruct" inputs for image A, whose triple features are synthesized by remapping from other images (B, C, and D), compelling the model to generate purer triple features.

In summary, our main contributions are as follows:

- We propose **TripleFDS**, a novel framework combining explicit disentanglement and synthesis. It achieves accurate feature disentanglement through a self-supervised regularization strategy and ensures robust synthesis via feature remapping.
- We introduce a new dataset, **SCB Synthesis**. By leveraging SCB Groups, it enhances training data diversity and facilitates robust disentanglement.
- **TripleFDS** achieves state-of-the-art performance on mainstream STE benchmarks, while enabling flexible triple feature combinations for new operations like style replacement and background transfer.

2 Related Work

Existing techniques for Scene Text Editing (STE) can be broadly categorized based on their approach to feature disentanglement: non-disentangling/implicit editing, and explicit disentanglement.

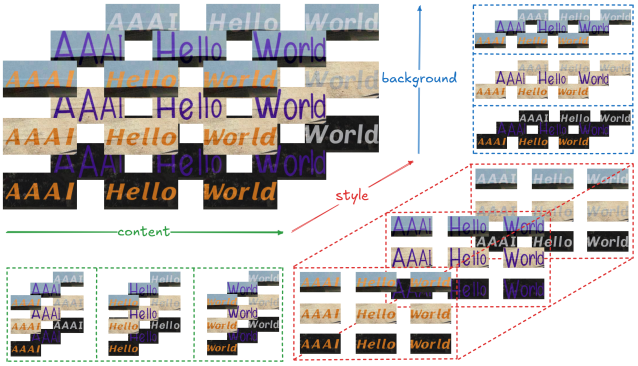

Figure 2: Visualizing $3 \times 3 \times 3$ SCB Group and Feature Disentanglement.

2.1 Non-disentangling or Implicit Editing Methods

These methods primarily focus on text reconstruction or inpainting, without explicitly disentangling the aftermentioned triple features.

Diffusion-inpainting-based methods, such as TextDiffuser (Chen et al. 2023b), AnyText (Tuo et al. 2024), Brush Your Text (Zhang et al. 2024b), UDiffText (Zhao and Lian 2024), DreamText (Wang et al. 2025), and Type-R (Shimoda et al. 2025), perform local text erasure and masked region filling. Despite advancements in quality, their implicit handling of features often leads to stylistic entanglement, which results in blurriness caused by residuals.

Transformer-decoder-based methods, such as DARLING (Zhang et al. 2024a) and RS-STE (Fang et al. 2025), leverage sequence-to-sequence generation and attention(Vaswani et al. 2017) mechanisms for text editing. While achieving text consistency and precise content editing, these methods also suffer from insufficient feature decomposition (implicit editing), leading to artifacts and limited editing capabilities, such as the inability to independently transfer style or background.

2.2 Explicit Disentanglement Methods

Explicit feature disentanglement aims to enhance the quality and flexibility of STE by separating visual components.

Early STE research, primarily based on Generative Adversarial Networks (GANs)(Goodfellow et al. 2014), often utilized distinct modules for tasks such as text conversion, background inpainting, style extraction, and foreground-background fusion(Shu et al. 2024). This modular approach represented an early form of explicit disentanglement. Methods like SRNet (Wu et al. 2019), SwapText (Yang, Huang, and Lin 2020), TextStyleBrush (Krishnan et al. 2023), and MOSTEL (Qu et al. 2023b) focused on tasks such as word or text-line editing, content replacement, and aesthetic transfer. Despite this modularity, their generalization ability was limited by the inherent capacity constraint of GANs and the difficulty of accurately decomposing text styles, leading to unstable background recovery and undesirable fusion artifacts (Shu et al. 2024).

Diffusion models, particularly in the cropping-reconstruction paradigm, also enable explicit disentanglement. SceneVTG (Zhu et al. 2024) employs MLLM reasoning for text generation, TextCtrl (Zeng et al. 2024) controls through style and glyph disentanglement, and FontDiffuser (Yang et al. 2024) focuses on one-shot font generation for diverse styles. However, TextCtrl's binary disentanglement (coupling style and background) often results in stylistic entanglement and unreliable background guidance from unmasked regions.

Our **TripleFDS** framework advances previous disentanglement methods. Unlike binary disentanglement, which entangles style and background, our approach achieves precise disentanglement of triple features, resulting in truly independent features. This robust disentanglement enables flexible combinations and diverse STE operations, addressing the stylistic entanglement and boundary issues faced by previous methods.

3 Method

This section details our proposed Scene Text Editing (STE) framework, **TripleFDS**, and its supporting dataset **SCB Synthesis**. Given that **TripleFDS** relies on the SCB Group as a fundamental data unit, we first introduce the novel dataset construction paradigm in Section 3.1. Subsequently, Section 3.2 provides an overview of the **TripleFDS** framework, explaining its disentanglement and synthesis processes. In Section 3.3, we present the key feature disentanglement constraints, which ensure semantic accuracy and non-redundancy of features. Finally, Section 3.4 discusses our feature remapping strategy, a crucial component for robust feature learning and preventing model collapse.

3.1 SCB Dataset Construction Paradigm

We introduce a systematic method for constructing text image datasets by explicitly decomposing complex scene text images into three independent dimensions: *background*, *content*, and *style*. Background refers to image regions outside the text mask, content denotes semantic characters within the text mask, and style encompasses visual features inside the text mask, such as font color, texture, glyph, border, and graphic transformations.

As depicted in Fig. 2, we propose the SCB Group concept. An SCB Group is an image collection formed by systematically combining information from these dimensions. An STE image I is modeled as $I = \mathcal{G}(S, C, B)$. For example, a minimal SCB Group comprises 8 text images derived from pairwise combinations of two styles S_x, two contents C_y, and two backgrounds B_z, resulting in $2 \times 2 \times 2 = 8$ variations, expressed as $\{I_{S_x C_y B_z} \mid x, y, z \in \{1, 2\}\}$. With the SCB Group, we can not only construct editing pairs for various features (e.g., content editing, style switching, and background changing), but also leverage its inherent structure, specifically the fixed feature correspondence among samples, as depicted in Fig. 3 (top-left). This enables robust training via the feature remapping strategy (Section 3.4) and fosters accurate feature disentanglement through contrastive learning with positive/negative sample pairing (Section 3.3).

To mitigate the significant domain gap observed between synthetic and real datasets in prior methods, we synthesize data with enhanced realism. The detailed methodologies for foreground text and style processing, including font selection, color configuration, and graphic transformations, as well as the strategies for foreground and background fusion, are thoroughly described in Appendix B.

3.2 Overall Framework

Our framework **TripleFDS** consists of two key processes: feature disentanglement and feature synthesis. Before describing these processes, we define the key notations used throughout this section: the VAE(Kingma and Welling 2014) sampling rate is denoted as f, the hidden dimension is d, and the character vocabulary size is $|\Sigma|$. The image embedding sequence length is $N = (H \times W)/f^2$ (where H and W represent the image height and width), and the text embedding sequence length is L (representing the maximum input character count).

During training, we first perform feature disentanglement to extract the core triple features, followed by feature synthesis to reconstruct the image.

1. **Feature Disentanglement.** The first process involves disentangling the input image into three key features: content, style, and background. This is achieved using a Transformer-decoder-based disentanglement module $\mathcal{F}_{\text{dis}}$ (Radford et al. 2019). The input image embedding $E_{I_{\text{src}}} \in \mathbb{R}^{N \times d}$ is obtained via a VAE encoder $\mathcal{E}$, while learnable query tokens ($Q_C \in \mathbb{R}^{L \times d}$, $Q_S \in \mathbb{R}^{(N-L) \times d}$, $Q_B \in \mathbb{R}^{N \times d}$) guide the extraction of the content, style, and background features. The disentanglement operation produces the following features:
$$[E_{\text{ignore}}, E_{C_{\text{src}}}, E_{S_{\text{src}}}, E_{B_{\text{src}}}] = \mathcal{F}_{\text{dis}}([E_{I_{\text{src}}}, Q_C, Q_S, Q_B]), \tag{1}$$
where $E_{C_{\text{src}}} \in \mathbb{R}^{L \times d}$ represents the content feature, $E_{S_{\text{src}}} \in \mathbb{R}^{(N-L) \times d}$ represents the style feature, and $E_{B_{\text{src}}} \in \mathbb{R}^{N \times d}$ represents the background feature. E_{ignore} serves as a placeholder. These disentangled features are further constrained using the self-supervised regularization loss from Section 3.3, ensuring semantic accuracy and non-redundancy.
2. **Feature Synthesis.** In the second process, the disentangled background and style features ($E_{B_{\text{src}}}$, $E_{S_{\text{src}}}$) are combined with the text embedding $E_{T_{\text{src}}} \in \mathbb{R}^{L \times d}$, derived from a character embedding lookup table $E_{\text{char}} \in \mathbb{R}^{|\Sigma| \times d}$, and an image reconstruction query $Q_I \in \mathbb{R}^{N \times d}$. This process uses another Transformer-decoder-based module $\mathcal{F}_{\text{synth}}$ for feature synthesis. Before synthesis, we apply the feature remapping strategy from Section 3.4 to create a "hard-to-reconstruct" triplet, ensuring robust feature learning. The synthesis operation is defined as:
$$[E_{\text{ignore}}, E_{I_{\text{rec}}}] = \mathcal{F}_{\text{synth}}([E_{B_{\text{src}}}, E_{S_{\text{src}}}, E_{T_{\text{src}}}, Q_I]). \tag{2}$$
In this process, $E_{I_{\text{rec}}}$ represents the embedding of the reconstructed image. Notably, instead of using the content feature $E_{C_{\text{src}}}$, the module leverages the text embedding $E_{T_{\text{src}}}$ to guide content generation, ensuring accurate and pure content guidance during both training and inference.

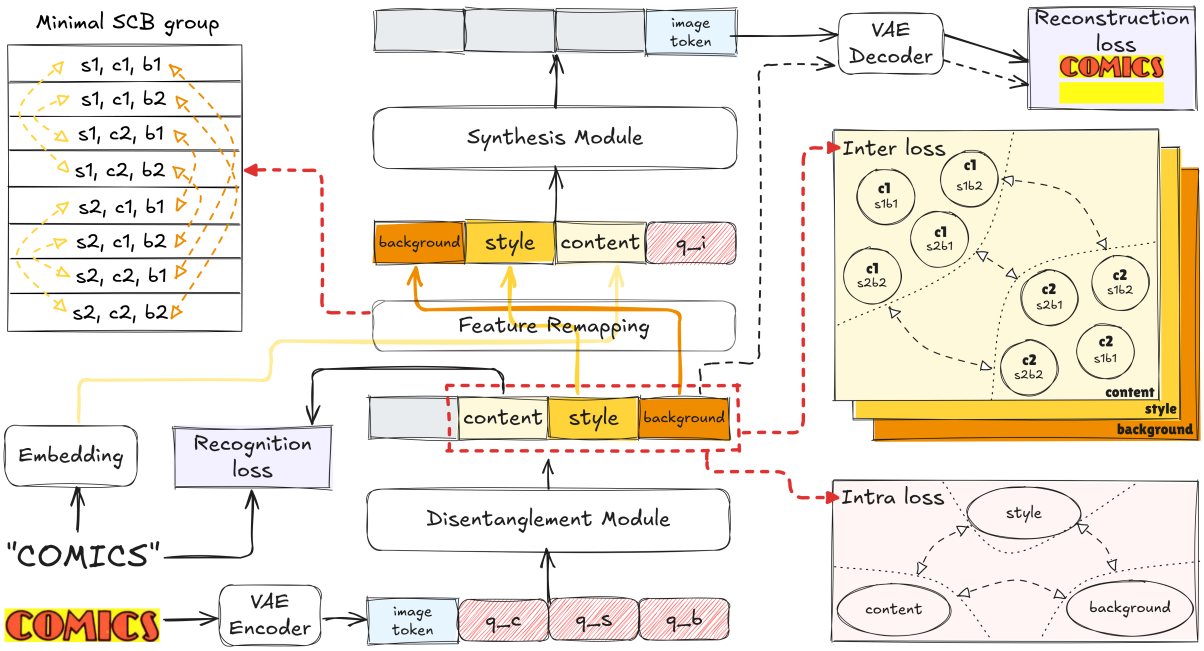

Figure 3: Overview of our **TripleFDS** framework. This figure illustrates its core components and strategies: a minimal SCB Group (top-left) where yellow and brown lines denote remapping objects for style and background features under the *Remapping Strategy*; the pipeline for feature disentanglement and synthesis (middle), with rounded rectangles representing network structures, red diagonal lines indicating learnable token; and visualizations of the *Inter loss* and *Intra loss* (bottom-right).

During inference, we apply the same framework as in training, but with modified inputs depending on the task. For text editing, we replace the input text with the target text. For style or background transfer tasks, we use the source image and condition image as input. The relevant feature from the condition image is used to replace the corresponding feature in the source image before the synthesis process.

3.3 Feature Disentanglement Constraints

Feature disentanglement is crucial for stable reconstruction and robust feature learning, its efficacy depends critically on the purity of the features from the disentanglement process. If the disentanglement process produces under-disentangled or redundant representations, it will severely hinder both the learning process and reconstruction accuracy.

Therefore, precise feature disentanglement is achieved via tailored contrastive loss functions. Firstly, the *inter-group contrastive loss* $\mathcal{L}_{inter}$ is employed to ensure that each feature (background, style, content) accurately corresponds to its respective image component and semantic token, thereby facilitating effective feature disentanglement across different samples. As shown in Fig. 3 (right-middle), in a minimal $2 \times 2 \times 2$ SCB Group, content feature learning defines positive and negative samples based on shared content. This concept extends across the batch, incorporating samples from other SCB Groups as additional negative samples to increase learning difficulty and foster training stability. High-dimensional features are projected for dimensionality reduction and then normalized to enable cosine similarity computation for contrastive learning, and an improved multi-round InfoNCE(He et al. 2020) loss optimizes style, content, and background features by adjusting positive/negative sample partitioning. The InfoNCE loss is:

$$\mathcal{L}_{\text{inter}_{i,j}} = -\log \frac{e^{sim(i,j)/\tau}}{e^{sim(i,j)/\tau} + \sum_{k \notin (\{i\} \cup P_i)} e^{sim(i,k)/\tau} + \epsilon}. \tag{3}$$

Here, P_i denotes the set of indices of all positive samples for anchor i (excluding i itself). The term $sim(i,j)$ denotes the cosine similarity between projected feature vectors F_i and F_j of anchor i and sample j. τ is a learnable temperature parameter, and ϵ is a small constant for numerical stability. The final $\mathcal{L}_{\text{inter}}$ loss is computed by averaging all such $\mathcal{L}_{\text{inter}_{i,j}}$ terms across all samples and all triple feature types (background, style, and content), aiming to maximize similarity between anchors and their positive samples while minimizing similarity with all negative samples.

To obtain purer disentangled features and mitigate implicit intra-sample coupling, we employ an *intra-sample multi-feature similarity loss* $\mathcal{L}_{\text{intra}}$. This loss calculates the cosine similarity between the features of the projected background B_i, the style S_i, and the content C_i for each sample. The similarity loss is defined as:

$$\mathcal{L}_{\text{intra}} = \frac{1}{3} \left(|\text{sim}(B_i, S_i)| + |\text{sim}(B_i, C_i)| + |\text{sim}(S_i, C_i)| \right). \tag{4}$$

Minimizing these similarity scores forces orthogonality in the latent space, thereby eliminating redundancy, as shown in Fig. 3 (bottom-right).

3.4 Feature Remapping Strategy

As described in Section 3.1, the SCB Group facilitates diverse STE tasks by combining styles, contents, and backgrounds. While this enables various feature editing pairs, designing separate tasks for every permutation becomes overly complex. All tasks fundamentally involve feature disentanglement and recombination, differing primarily in how features are grouped. Therefore, instead of traditional editing-centric training (Qu et al. 2023b; Zeng et al. 2024; Fang et al. 2025), we adopt a source image reconstruction approach: disentangling an image into its triple features and reconstructing the original image using them, as shown in Fig. 3 (middle).

A challenge in reconstruction training is that one feature (e.g., background) may carry redundant information, allowing the model to "shortcut", leading to collapse where content and style features are minimally learned. To prevent this, we introduce a *feature remapping strategy*. This strategy uses the fixed feature mapping relationships in the SCB Group to remap features during reconstruction, forcing the model to rely on the correct feature combinations.

After decomposing the original image into text style, text content, and background features, remapped feature triplets are created by leveraging SCB Group mappings. The remapping logic is as shown in Fig. 3 (top-left).

- The **background feature** is remapped with another background from the same SCB Group, ensuring background identity while varying content and style (brown dotted line).
- The **style feature** is remapped with another style from the same SCB Group, maintaining style identity but varying background and content (yellow dotted line).
- The **content feature** is not remapped, as it is derived directly from the input text embedding, ensuring consistency across images with the same text content.

If multiple valid remapping samples exist for a feature, their average forms the final remapped feature.

For the background feature, the remapping formula is:

$$\bar{B}_{\text{remap}} = \frac{1}{(N_S-1)(N_C-1)} \sum_{\substack{s'\in\{1,\dots,N_S\}\\ c'\in\{1,\dots,N_C\}\\ s'\neq s, c'\neq c}} B|I_{s',c',b}. \quad (5)$$

Here, $\bar{B}_{\text{remap}}$ is the new background feature for the current sample $I_{s,c,b}$, and $B|I_{s',c',b}$ is the background feature from image $I_{s',c',b}$ in the SCB Group, averaged over styles s' and contents c'.

The hybrid triplet is fed into the feature synthesis module, which must reconstruct the original image. This targeted interference forces the module to rely only on the target image features, penalizing redundant encoding (e.g., style/content in the background) by causing reconstruction failure and higher loss. This mechanism encourages the disentanglement module to learn purer features, ensuring that each feature type (background, style) captures only its relevant information, creating "hard-to-reconstruct" inputs and establishing an information bottleneck that minimizes inter-feature correlation.

In addition to the self-supervised loss discussed in Section 3.3, the availability of ground truth labels for content and background features allows us to apply supervised constraints. Following the design of RS-STE (Fang et al. 2025), we introduce two loss functions: a reconstruction loss $\mathcal{L}_{\text{rec}}$ for the quality of the inpainting background and the reconstruction image and a recognition loss $\mathcal{L}_{\text{reg}}$ for the accuracy of content feature.

The total loss function is then defined as:

$$\mathcal{L}_{\text{total}} = (\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{reg}}) + \lambda_{\text{inter}}\mathcal{L}_{\text{inter}} + \lambda_{\text{intra}}\mathcal{L}_{\text{intra}}. \quad (6)$$

Here, $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{reg}}$ are the baseline losses from RS-STE (Fang et al. 2025), while $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$ are new loss terms introduced for our model, with their respective settings of weights are provided in Appendix C.

4 Experiments

4.1 Datasets and Evaluation Metrics

Training Data. Our model is trained exclusively on synthetic datasets. During training, we utilized a 1 million-image **SCB Synthesis** dataset, generated by our dataset construction paradigm (Section 3.1). For a fair evaluation on Tamper-Syn2k(Qu et al. 2023b), our model was further fine-tuned on MOSTEL's synthetic dataset, Tamper-train-150k.

Evaluation Data. Model performance was comprehensively assessed on various real-world and synthetic datasets, including the ScenePair(Zeng et al. 2024) benchmark (1280 real-world editing-pairs), the Tamper-Syn2k(Qu et al. 2023b) dataset (2,000 synthetic editing-pairs), and the Tamper-Scene(Qu et al. 2023b) dataset (7,725 unpaired real-world images).

Evaluation Metrics. For visual quality, we adopt: (i) Mean Squared Error (MSE) for pixel difference; (ii) Peak Signal-to-Noise Ratio (PSNR) for signal reconstruction quality; (iii) Structural Similarity Index Measure (SSIM) for structural similarity; and (iv) Fréchet Inception Distance (FID)(Heusel et al. 2017) for realism and diversity (via feature distribution comparison). For text rendering accuracy, we measure word accuracy (ACC), assessing character-level correctness within edited images.

Comparison. We conduct a quantitative comparison of our **TripleFDS** against several methods in STE. These include GAN-based (SRNet (Wu et al. 2019), MOSTEL (Qu et al. 2023b)), Diffusion-based (AnyText (Tuo et al. 2024), TextCtrl (Zeng et al. 2024)), and Transformer-based (DARLING (Zhang et al. 2024a), RS-STE (Fang et al. 2025)) methods. As shown in Tab. 1, certain entries are dashed due to specific limitations: AnyText lacks results for Tamper-Syn2k and Tamper-Scene as it is designed for inpainting with cropped inputs. DARLING's ScenePair results are unavailable because its code is not open-source, and its paper only reports performance on Tamper-Syn2k and Tamper-Scene. For a fair comparison, we implement RS-STE (denoted RS-STE† in Tab. 1). This was necessitated by RS-STE's use of MLT2017(Nayef et al. 2017) for fine-tuning,

Figure 4: Comparison of previous methods with ours. Previous methods tend to generate incorrect or fused text, as shown in the red boxes, while **TripleFDS** effectively mitigates these problems.

Methods	Tamper-Scene	Tamper-Syn2k				ScenePair				
	ACC↑	SSIM↑	PSNR↑	MSE↓	FID↓	ACC↑	SSIM↑	PSNR↑	MSE↓	FID↓
SRNet	30.26	49.97	18.66	2.16	64.37	17.84	26.66	14.08	5.61	49.22
MOSTEL	66.54	56.94	20.27	1.35	33.79	37.69	27.45	14.46	5.19	49.19
AnyText	–	–	–	–	–	51.12	30.73	13.66	6.19	51.79
DARLING	70.85	60.07	20.80	1.20	44.48	–	–	–	–	–
TextCtrl	74.17	66.60	20.79	1.30	31.13	84.67	37.56	14.99	4.47	**43.78**
RS-STE†	73.71	65.91	20.92	**1.17**	36.49	89.92	42.59	15.71	3.56	48.82
Ours	**75.62**	**67.44**	**21.60**	1.26	**30.84**	**93.58**	**44.54**	**16.53**	**3.23**	46.40

Table 1: Comparison on editing performance with previous methods on Tamper-Syn2k, Tamper-Scene and ScenePair. The MSE and SSIM are presented as $(\times 10^{-2})$, and RecAcc is presented in percent $(\%)$.

which could cause data leakage from ScenePair. Our re-implementation, based on their paper and code, ensured fairness by scaling up its parameter count to match ours and adapting our SCB Group dataset for its pretraining.

4.2 Quantitative and Qualitative Analysis

As shown in Tab. 1, **TripleFDS** achieves state-of-the-art performance across mainstream STE benchmarks.

For image quality, **TripleFDS** demonstrates strong performance. Specifically, on Tamper-Syn2k, we achieved leading performance in SSIM (**67.44**), PSNR (**21.60**), and FID (**30.84**). These results indicate that our disentanglement method successfully yields more accurate glyph structures, purer style features, and higher-quality background reconstruction. For MSE, the "hard-to-reconstruct" triplets employed by the remapping strategy in the synthesis stage, entangled with potential abnormal pixels in synthetic data, can pose challenges to the model, sometimes causing confusion during feature synthesis. On ScenePair, we secured leading performance in SSIM (**44.54**), PSNR (**16.53**), and MSE (**3.23**). **TripleFDS** achieved the second-best FID of **46.40**, only behind TextCtrl (FID: **43.78**), which excels in real-scene background inpainting due to its powerful diffusion architecture. Compared to RS-STE†(FID: **48.82**), **TripleFDS** significantly improves performance, underscoring the superior effectiveness of explicit disentanglement over implicit editing for high-quality, realistic image generation.

Regarding text accuracy, **TripleFDS** achieves the highest accuracy on both Tamper-Scene (**75.62%**) and ScenePair (**93.58%**), surpassing baselines like TextCtrl and RS-STE†. Despite RS-STE†also employing a recognition loss to emphasize the accuracy of the text, our triple feature disentanglement purifies the style and background features by reducing redundant content information, thus minimizing the appearance of artifacts and improving the accuracy of text recognition.

Upon closer inspection of Fig. 4, **TripleFDS** demonstrates superior editing, effectively avoiding artifacts and color aberrations while maintaining strong style consistency.

4.3 Ablation

We conducted comprehensive ablation experiments to systematically evaluate the contribution of our key technical designs, hyperparameter choices, and model capacity and training data scale to model performance on the ScenePair dataset.

The following subsections detail experiments on core method designs, SCB Group configurations, and different editing operations.

Figure 5: Different editing operations of **TripleFDS**, with the operations highlighted in purple representing those that **TripleFDS** can perform in addition to the capabilities of previous methods.

Methods	ACC↑	SSIM↑	PSNR↑	FID↓
($\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{reg}}$)	90.23	41.84	15.48	49.31
+ RS	92.24	43.18	16.01	46.56
+ $\mathcal{L}_{\text{inter}}$	93.14	**44.73**	16.49	47.08
+ $\mathcal{L}_{\text{intra}}$ (Ours)	**93.58**	44.54	**16.53**	**46.40**

Table 2: Ablation results for core method design.

Ablation of core method designs. As depicted in Tab. 2, our foundational model was trained solely with reconstruction loss $\mathcal{L}_{\text{rec}}$ and recognition loss $\mathcal{L}_{\text{reg}}$.

Introducing the feature remapping strategy RS was crucial, effectively preventing model collapse and improving FID (49.31 to 46.56) and other metrics. This efficacy stems from its implicit diverse feature remapping via "hard-to-reconstruct" triplets, fully leveraging SCB Group's inherent structure for robust feature learning.

Adding the inter-group contrastive loss $\mathcal{L}_{\textbf{inter}}$ further enhanced performance in ACC, SSIM, and PSNR. Its emphasis on regional feature focus, while maintaining structural integrity, subtly increased FID (46.56 to 47.08) due to residual background feature redundancy interfering with detail recovery.

Finally, incorporating the intra-sample multi-feature similarity loss $\mathcal{L}_{\textbf{intra}}$ completed our full model. This orthogonality constraint directly mitigated redundancy, further improving FID. However, it marginally impacted overall feature expressiveness, resulting in a minor SSIM drop.

Ablation of different SCB Group configurations. This ablation study investigates the impact of SCB Group's internal structure and diversity distribution on disentanglement effectiveness. As detailed in Tab. 3, we investigated various SCB Group configurations (e.g., (1,4,4,4), (4,4,2,2), (4,2,4,2), (4,2,2,4), (8,2,2,2)), each totaling a 64-image batch, exploring distributions of groups, styles, contents, and backgrounds.

Results show that biasing a single feature dimension degrades overall performance. For example, (4,2,4,2) showed notably worse image quality (FID: 49.48), despite increased content diversity. More balanced configurations like (1,4,4,4) and especially our standard (8,2,2,2) yielded better overall results.

Configuration	ACC↑	SSIM↑	PSNR↑	FID↓
(1,4,4,4)	93.13	43.53	15.86	47.52
(4,4,2,2)	90.89	42.17	15.76	47.59
(4,2,4,2)	92.2	41.95	15.63	49.48
(4,2,2,4)	90.56	42.63	15.7	46.79
(8,2,2,2)	**93.58**	**44.54**	**16.53**	**46.40**

Table 3: Ablation results for training data configuration. (Number of groups, styles per group, content per group, background per group) indicating image quantity combination, totaling 64 images per batch.

The optimal performance of (8,2,2,2) stems from increased negative sample diversity for contrastive learning, benefiting from a larger number of groups. Conversely, when feature dimensions exceed two (e.g., in (1,4,4,4)), averaging multiple remapping objects in the feature remapping strategy can significantly increase training difficulty due to limited model capacity, thus favoring more groups with fewer elements per dimension.

Qualitative Analysis of Various Text Editing Operations. Fig. 5 presents the visualization results of **TripleFDS** across various editing operations. **TripleFDS** successfully disentangles text, style, and background, offering a wider array of possibilities for scene text editing.

5 Conclusion

To overcome challenges in feature disentanglement in prior scene text editing (STE) methods, we introduce **TripleFDS**, a novel framework with disentangled modular attributes, alongside the **SCB Synthesis** dataset. This dataset provides robust training data for triple feature disentanglement using the SCB Group. Leveraging this construct as a foundational unit, **TripleFDS** first disentangles the triple features, ensuring semantic accuracy by applying inter-group contrastive regularization and minimizing feature redundancy. During synthesis, **TripleFDS** remaps features to prevent "shortcut" phenomena and avoid feature leakage in reconstruction. Extensive experiments show that **TripleFDS** outperforms existing methods, offering enhanced flexibility and high-quality results, achieving sota performance in STE.

Acknowledgments

This work is supported in part by National Natural Science Foundation of China (Grant No. 62276121), the TianYuan funds for Mathematics of the National Science Foundation of China (Grant No. 12326604).

References

Chen, H.; Xu, Z.; Gu, Z.; Lan, J.; Zheng, X.; Li, Y.; Meng, C.; Zhu, H.; and Wang, W. 2023a. DiffUTE: Universal Text Editing Diffusion Model. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Chen, J.; Huang, Y.; Lv, T.; Cui, L.; Chen, Q.; and Wei, F. 2023b. TextDiffuser: Diffusion Models as Text Painters. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Fang, Z.; Lyu, P.; Wu, J.; Zhang, C.; Yu, J.; Lu, G.; and Pei, W. 2025. Recognition-Synergistic Scene Text Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13104–13113.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. arXiv:1406.2661.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic Data for Text Localisation in Natural Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2315–2324.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9726–9735.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

Ji, J.; Zhang, G.; Wang, Z.; Hou, B.; Zhang, Z.; Price, B. L.; and Chang, S. 2024. Improving Diffusion Models for Scene Text Editing with Dual Encoders. *Transactions on Machine Learning Research*, 2024.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*.

Krishnan, P.; Kovvuri, R.; Pang, G.; Vassilev, B.; and Hassner, T. 2023. TextStyleBrush: Transfer of Text Aesthetics From a Single Example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 9122–9134.

Liu, Z.; Liang, W.; Liang, Z.; Luo, C.; Li, J.; Huang, G.; and Yuan, Y. 2024. Glyph-ByT5: A Customized Text Encoder for Accurate Visual Text Rendering. In *European Conference on Computer Vision (ECCV)*, 361–377.

Luo, D.; Liu, Y.; Yang, R.; Liu, X.; Zeng, J.; Zhou, Y.; and Bai, X. 2025. Toward real text manipulation detection: New dataset and new solution. *Pattern Recognition*, 157: 110828.

Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; Khlif, W.; Luqman, M. M.; Burie, J.; Liu, C.; and Ogier, J. 2017. ICDAR2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification - RRC-MLT. In *International Conference on Document Analysis and Recognition (ICDAR)*, 1454–1459.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H. V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.; Li, S.; Misra, I.; Rabbat, M.; Sharma, V.; Synnaeve, G.; Xu, H.; Jégou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024.

Qu, C.; Liu, C.; Liu, Y.; Chen, X.; Peng, D.; Guo, F.; and Jin, L. 2023a. Towards Robust Tampered Text Detection in Document Image: New Dataset and New Solution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5937–5946.

Qu, C.; Zhong, Y.; Guo, F.; and Jin, L. 2024. Generalized Tampered Scene Text Detection in the era of Generative AI. *CoRR*, abs/2407.21422.

Qu, Y.; Tan, Q.; Xie, H.; Xu, J.; Wang, Y.; and Zhang, Y. 2023b. Exploring Stroke-Level Modifications for Scene Text Editing. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*, 2119–2127.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multimodal Learners. *OpenAI Blog*. Accessed: 2024-11-15.

Roy, P.; Bhattacharya, S.; Ghosh, S.; and Pal, U. 2020. STEFANN: Scene Text Editor Using Font Adaptive Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13225–13234.

Shimoda, W.; Inoue, N.; Haraguchi, D.; Mitani, H.; Uchida, S.; and Yamaguchi, K. 2025. Type-R: Automatically Retouching Typos for Text-to-Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2745–2754.

Shu, Y.; Zeng, W.; Li, Z.; Zhao, F.; and Zhou, Y. 2024. Visual Text Meets Low-level Vision: A Comprehensive Survey on Visual Text Processing. *CoRR*, abs/2402.03082.

Tuo, Y.; Geng, Y.; and Bo, L. 2024. AnyText2: Visual Text Generation and Editing With Customizable Attributes. *CoRR*, abs/2411.15245.

Tuo, Y.; Xiang, W.; He, J.; Geng, Y.; and Xie, X. 2024. AnyText: Multilingual Visual Text Generation and Editing. In *International Conference on Learning Representations (ICLR)*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6000–6010.

Wang, A.; Wang, J.; Yan, Z.; Shang, W.; Lin, R.; and Zhang, Z. 2024. TextMaster: Universal Controllable Text Edit. *CoRR*, abs/2410.09879.

Wang, Y.; Zhang, W.; Xu, H.; and Jin, C. 2025. DreamText: High Fidelity Scene Text Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 28555–28563.

Wu, L.; Zhang, C.; Liu, J.; Han, J.; Liu, J.; Ding, E.; and Bai, X. 2019. Editing Text in the Wild. In *Proceedings of the 27th ACM International Conference on Multimedia (ACM MM)*, 1500–1508.

Yang, Q.; Huang, J.; and Lin, W. 2020. SwapText: Image Based Texts Transfer in Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14688–14697.

Yang, Y.; Gui, D.; Yuan, Y.; Liang, W.; Ding, H.; Hu, H.; and Chen, K. 2023. GlyphControl: Glyph Conditional Control for Visual Text Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, Z.; Peng, D.; Kong, Y.; Zhang, Y.; Yao, C.; and Jin, L. 2024. FontDiffuser: One-Shot Font Generation via Denoising Diffusion with Multi-Scale Content Aggregation and Style Contrastive Learning. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, 6603–6611.

Ye, H.; Zhang, J.; Liu, S.; Han, X.; and Yang, W. 2023. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. *CoRR*, abs/2308.06721.

Zeng, W.; Shu, Y.; Li, Z.; Yang, D.; and Zhou, Y. 2024. TextCtrl: Diffusion-based Scene Text Editing with Prior Guidance Control. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhang, B.; Xie, H.; Gao, Z.; and Wang, Y. 2024a. Choose What You Need: Disentangled Representation Learning for Scene Text Recognition, Removal and Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 28358–28368.

Zhang, L.; Chen, X.; Wang, Y.; Lu, Y.; and Qiao, Y. 2024b. Brush Your Text: Synthesize Any Scene Text on Images via Diffusion Model. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, 7215–7223.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3813–3824.

Zhao, L.; Chen, C.; and Huang, J. 2021. Deep Learning-Based Forgery Attack on Document Images. *IEEE Transactions on Image Processing*, 30: 7964–7979.

Zhao, Y.; and Lian, Z. 2024. UDiffText: A Unified Framework for High-Quality Text Synthesis in Arbitrary Images via Character-Aware Diffusion Models. In *European Conference on Computer Vision (ECCV)*, 217–233.

Zhu, Y.; Liu, J.; Gao, F.; Liu, W.; Wang, X.; Wang, P.; Huang, F.; Yao, C.; and Yang, Z. 2024. Visual Text Generation in the Wild. In *European Conference on Computer Vision (ECCV)*, 89–106.